Distribution of Dockerfiles: Who is responsible for FOSS Licence Compliance?

Dr. Till Jaeger[a]

(a) Partner, JBB Rechtsanwälte

Abstract

The deployment of Docker is becoming increasingly popular as container technology allows for a unified software distribution that is largely independent of the target system. This raises new questions of FOSS license compliance. The reason is that in addition to the complete software distribution as a "Docker image", so-called Dockerfiles can be used, which - similar to a script - contain a kind of construction manual for the software which may include downloads from third party repositories. Such form of decentralized distribution raises the question of responsibility for compliance with the license conditions. This article sheds light on the concept of "distribution" under European copyright law as a starting point for the interpretation of free licenses. In the course of the study, it is shown that physical distribution and distribution in the meaning of copyright law do not always have to be congruent.

This article was created in collaboration with and funded by the Open Source Automation Development Lab (OSADL) eG (www.osadl.org).

Keywords

Law; information technology; Free and Open Source Software (FOSS); license compliance; Docker

1. Introduction and problem

Docker technology and the associated FOSS licensing compliance issues have become a focus of research in recent years. In particular, reference should be made to the extensive analysis by Armijn Hemel, who described the technical foundations of the Docker technology and raised the relevant license compliance issues in the white paper "Docker Containers for Legal Professionals".[1] Hemel drew attention in particular to the open question of who is responsible for ensuring that the licensing conditions for the software components are complied with, when an acquirer of the Dockerfile downloads that file from a third-party source.

---

1   Hemel, Armijn, (2020), 'Docker Containers for Legal Professionals,' [pdf] Available at: <https://www.linuxfoundation.org/wp-content/uploads/Docker-Containers-for-Legal-Professionals-Whitepaper_042420.pdf> [Accessed 16 February 2021]. See also Peterson, Scott, (2020), 'Making compliance scalable in a container world.' Available at: <https://opensource.com/article/20/7/compliance-containers> [Accessed 16 February 2021].

Almost all licenses for FOSS tie the compliance with license obligations to the "distribution" (or similar language, such as "conveying" in GPLv3) of the software. In most cases, the terms "distribution" and "conveying" are not further defined and refer to the applicable copyright law.[2]

Because of its great importance for license compliance, the term "distribution" has repeatedly been the subject of legal analysis. In particular, the enlightening article by Heather Meeker in this journal, written from the perspective of US copyright, is worth mentioning.[3] Although many open source licenses have been drafted against the background of US copyright law, it is to be expected that European courts will fall back on the understanding of the term "distribution" as elaborated by the Court of Justice of the European Union (CJEU).

In the course of the article, we will first provide an overview of the technical basics of Docker as well as the interpretation of the term "distribution" under European copyright law. This is followed by a discussion of who is responsible for license compliance when distributing Dockerfiles.

2. Technical background of the Docker technology

Docker is a technology for the installation and deployment of programs in a container. It has the advantage that all dependencies are present in one technical unit and are largely independent of the host system. Unlike virtualization via Hypervisor, a Docker container does not contain an operating system kernel; instead, a specific operating system command makes the filesystem tree of the container appear as root directory for all programs of the container. Thus, the remaining filesystem outside the container becomes invisible for the container programs. Docker containers require a Unix-like operating system and are mainly intended to be used with a Linux kernel.

A pre-configured container can be distributed as a "Docker image" and often contains, in addition to the main program, an application, the dependencies as program code and, if required, utilities and configuration files. Docker images can be distributed individually but are in many cases accessible via public repositories such as "Docker Hub" as well. This is also true for so-called "base images" which contain essential system components such as C library, package manager, shell and a directory tree and which refer to a specific Linux distribution. On top of this base image, further functionalities can then be added as so-called "layers" that can be distributed separately as individual archive files but are built upon each other and form the complete Docker image.

A "Dockerfile" is a text file that - similar to a script - contains step-by-step instructions for setting up a Docker image. A Dockerfile can have its own license that usually only applies to the Dockerfile itself, but not to the programs included in the Docker container.

The "Docker engine" – a management software for Docker containers – can build Docker images with the help of Dockerfiles by sequentially processing the commands from the Dockerfile. Usually, individual components, for instance a base image and individual layers, are downloaded from internal or external repositories. This means that it is possible and, indeed, customary that a supplier provides a Dockerfile to their customers without passing on physical program code, and the customers can then build their own Docker container including program code, which completely or partially originates from public repositories.

This raises the question whether and which license obligations have to be complied with by the provider of the Dockerfile with regard to FOSS included in the Docker image that was built using the Dockerfile.

---

2 Sec. 0 GPL-3.0 provides as follows: "To 'convey" a work means any kind of propagation that enables other parties to make or receive copies." and "To 'propagate' a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy."
3 Meeker, Heather (2012), 'The Gift that Keeps on Giving – Distribution and Copyleft in Open Source Software Licenses', JOLTS, 4(1), pp 29 – 40, [DOI: 10.5033/ifosslr.v4i1.66].

3. Legal background – Distribution right under EU law

Almost all FOSS licenses make the requirement to comply with license obligations conditional on the act of distributing or conveying software as given by copyright law, i.e. as a rule, the passing of program copies of the program to third parties. Only few licenses – such as the GPL-3.0 for the term "convey" – include a definition of what is meant by "distribution". Thus, it is common practice to refer to the interpretation of the term in applicable copyright law. In Germany, this is the term "Verbreitung" (distribution) according to § 69c no 3 UrhG (German Copyright Act), which is described as "any form of distribution of the original of a computer program or of copies thereof, including rental."). In this case, "Verbreitung" is understood as in § 17 (1) UrhG which provides for this right of use for works other than computer programs

> *"The right of distribution is the right to offer the original or copies of the work to the public or to put it into circulation."*

and which is to be interpreted in the light of Article 4 of the Directive 2009/24/EG of the European Parliament and of the Council on the legal protection of computer programs.[4] The highest German and European courts, the German Federal Court of Justice (Bundesgerichtshof (BGH)) and the Court of Justice of the European Union (CJEU), have contributed to the interpretation of the right of distribution in numerous court decisions. This is discussed in more detail below.

4. Distribution of Dockerfiles – an analysis

The following sections will first examine the general question whether a physical transfer of the program code is required for a distribution under copyright law. Afterwards, there will be further explanations regarding the different components of a Docker image, namely base image, program libraries, patches and updates.

4.1 Requirement of physical distribution of program code?

"Distribution" can only be attributed to the provider of the Dockerfile in case that not only their own "physical" distribution of program copies is covered by the concept of distribution - as defined by copyright law - but also other acts that lead to the acquisition of a program copy from third-parties.

Here, it can first be stated that the highest courts in Germany and the EU regularly decide that not only the physical act is relevant but also other aspects are to be considered that could lead to a situation where the third party that physically performs the legally relevant act of copying or distributing is merely considered a "tool" of the one who appears as principal or initiator. These aspects include in particular the organizational control which is referred to by the CJEU as the "essential role".[5] One example is the "Internet radio recorder" decision of the German Federal Court of Justice (BGH), which dealt with whether fully automated recordings of digital radio stations by an Internet service are a (permissible) private copy by the client or an (unauthorized) copy by the service provider. In this case the BGH stated as follows:

> *"In this context, the decisive factor is whether the manufacturer is limited to 'taking the place of the reproduction device' and acting as a 'necessary tool' of the other party - in which case the reproduction is to be attributed to the purchaser - or whether he opens up a copyright-relevant use to an extent and intensity that cannot be reconciled with the considerations that justify the privileges of private use - then the reproduction is to be attributed to the manufacturer. Within the framework of this examination, which is based*

---

4   Directive 2009/24/EC on the legal protection of computer programs (codified version). Available at: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32009L0024> [Accessed 16 February 2021].
5   See the 'Opinion of Advocate General Saugmandsgaard Øe in the joined Cases C-682/18 and C-683/18 (Frank Peterson v Google LLC et al), ECLI:EU:C:2020:586. Available at: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:62018CC0682> [Accessed 16 February 2021].

> *on normative standards, it must also be determined whether the client has organizational sovereignty over the recording process."*[6]

The CJEU has based several decisions on who plays the "essential role" with respect to the act of exploitation of copyrights. This is especially apparent in § 17 UrhG (German Copyright Act) which designates as an act of distribution the mere "offer", thus, a preparatory act of a physical distribution. Although the European Directive does not explicitly mention the term "offer", the CJEU has decided as follows:

> *"Taking that context into account, the Court specifically found that distribution to the public is characterised by a series of acts going, at the very least, from the conclusion of a contract of sale to the performance thereof by delivery to a member of the public. A trader in such circumstances bears responsibility for any act carried out by him or on his behalf giving rise to a distribution to the public in a Member State where the goods distributed are protected by copyright. … As regards an invitation to submit an offer, or a non-binding advertisement for a protected object, those also fall under the series of acts taken with the objective of making a sale of that object. … In the light of the foregoing considerations, the answer to the questions referred is that Article 4(1) of Directive 2001/29 must be interpreted as meaning that it allows a holder of an exclusive right to distribute a protected work to prevent an offer for sale or a targeted advertisement of the original or a copy of that work, even if it is not established that that advertisement gave rise to the purchase of the protected work by an EU buyer, in so far as that that advertisement invites consumers of the Member State in which that work is protected by copyright to purchase it."*[7]

This and other decisions of the CJEU reveal that it does not, or not only, come down to distribution in a technical sense but that preparatory acts of distribution could be relevant as well, at least if the actor plays an "essential role" in the distribution process. This is precisely the case for Dockerfiles. The provider of the Dockerfile plays an essential role in the distribution of the software contained in the Docker image because the Dockerfile gives orchestrated instructions that - in accordance with their intended use - lead to the transmission of a complete functioning system at the recipient of the Dockerfile. In this respect, it is the provider who has the organizational control. Accordingly, the provider has to comply with the license obligations of FOSS distributed in this way.

The fact that the provider of the Dockerfile distributes the software referenced therein is not opposed by the fact that the operator of a repository, who offers the base image or layer for download, also performs an act of distribution of the program code or "makes it available to the public",[8] respectively. This is because mostly the respective layers are generally offered for download and not only with respect to a particular container. If the latter is the case, then the person or party that provides the layer through the repository potentially performs the act of distribution (or communication to the public respectively) and not the operator of the repository.

4.2 Patches

With additional layers even already installed programs can be modified. In that case, the docker container comprises the unmodified program in one layer and the modifications in another layer so that the modified program is being executed. In such situations, an "essential role" of the provider of the Dockerfile must always be assumed since the Dockerfile defines which modifications are applied. Accordingly, the license obligations must be complied with by the provider of the Dockerfile.

---

6   BGH (German Federal Court of Justice), judgment of 2020-03-05 - I ZR 32/19 – *Internet radio recorder*. Available at: <https://openjur.de/u/2202077.html> [Accessed 16 February 2021].
7   CJEU of 2015-05-13, C-516/13 – *Dimensione Direct Sales and Labianca*. Available at: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:62013CJ0516&qid=1607613372933&from=EN> [Accessed 16 February 2021].
8   Please not that the "Right of communication to the public of works and right of making available to the public" in Art. 3 are independent rights from the "distribution right" in Art. 4 Directive 2001/29/EC. Available at: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32001L0029> [Accessed 16 February 2021].

Attention should be paid to the fact that this applies for the original version of the program as well as for the modifications since both versions are distributed to the recipient (even though only the modified version is actually used).[9] The same applies when programs are removed by a new layer but are still physically included in the Docker image.

4.3 System requirements and base images

The starting point of the following considerations is that Open Source licenses do not (and cannot) extend to granting rights for the use of independent programs which do not fall within the scope of the license but may be required for deploying the Open Source software. A typical case is an operating system or a web server that is required for running an application. Such programs, which are independent but required for the execution of an application, are hereinafter referred to as "system requirements". Whoever distributes Dockerfiles is not responsible for complying with the license obligations of system requirements such as the Docker Engine or the Linux kernel; these are also not referenced in the Dockerfile.

The question is whether base images can also be considered to be system requirements. Programs included in a base image are generally independent of the application that is meant to be executed in a Docker container. As long as the programs that are included in the base image are used in their unmodified form, they can be considered to be system requirements without further ado since the fact that a Dockerfile contains download instructions does not necessarily make the provider of the Dockerfile also the provider of the base image. This is also reflected by the fact that where operator of the repository refuses access, a download is no longer possible. This is outside the control of the provider of the Dockerfile. The same is true in the case of a patch but there is, however, an additional criterion for treating patches and system requirements differently. It is typical for computer programs and distinguishes them from other copyrighted works that they are intended to work with other independent programs. For example, most applications do not run without an operating system. Accordingly, there is a need to install such system requirements but that does not mean that the provider of an application plays the essential role in the distribution.

The situation here is rather comparable with a download link. This leads to the question, fiercely debated in the EU, whether a download link to a copyrighted work constitutes an act relevant under copyright law, namely communicating it to the public (and therefore is possibly resulting in copyright infringement). Here, the CJEU has established a series of complex criteria which require a case-by-case assessment.[10] These criteria particularly assess the following questions: whether a new group of purchasers is being opened up; whether the intended use is for commercial purposes; whether the act plays a significant role for the offer; and whether the offer is illegitimate. A blanket statement is hardly possible here. It should be noted here that the consideration of these criteria was unusual in the member states of the EU and is probably due to the desire of the CJEU for greater harmonization of the legal situation of copyright on the Internet.

According to the view presented here, the operator of the repository and provider of the base image plays an essential role in the distribution of base images, whereas the reference in the Dockerfile rather facilitates the acquisition of system requirements. Therefore, the operator of the repository performs the act of communication to the public and must comply with the license obligations of the contained FOSS independently at least in cases where the offer is legitimate.

The interpretation stated above is the legal opinion of the author of this study; there is no case-law with regard to this specific context for computer programs in general, and Dockerfiles in particular. Different interpretations are certainly arguable (especially insofar that all referenced layers including

---

9   See Hemel Armijn, ibid n. 1, p. 19.
10   As the CJEU, judgment of 14 June 2017 in case C-610/15 – Stichting Brein (The Pirate Bay) itself declares: 'In order to determine whether a user is making a 'communication to the public' within the meaning of Article 3(1) of Directive 2001/29, it is necessary to take into account several complementary criteria, which are not autonomous and are interdependent. Consequently, those criteria must be applied both individually and in their interaction with one another, since they may, in different situations, be present to widely varying degrees." Available at: <http://curia.europa.eu/juris/liste.jsf?language=en&T,F&num=c-610-15> [Accessed 16 February 2021].

the base image are distributed – or communicated to the public respectively – by the provider of the Dockerfile).

It should be mentioned that, at present, numerous operators of repositories do not correctly comply with license obligations of FOSS licenses (e.g. they do not properly offer the source code of GPL and LGPL components) and are liable for copyright infringement. In this case, providing a Dockerfile that includes a reference to a license-violating offer can be considered as an independent act of distribution or at least as a contributory copyright infringement (i.e. an incitement to or aiding and abetting of license violation), if the provider of the Dockerfile has or should have knowledge of the license violation. It should therefore be reviewed whether the offer of the base image in the given repository is license compliant.[11] For recipients who want to use a Docker image internally, this is still permissible, since the mere program execution of FOSS is not restricted. For example, sec. 4 GPL-2.0 makes this clear.[12] However, if the recipient wants to redistribute Docker images, he/she must ensure compliance with the license conditions (see 4.6 below), because the distribution right is not exhausted in the case that the distribution of the Dockerfile is copyright infringing..

4.4 Program libraries

For libraries that are linked with a program, there is some disagreement as to whether these are considered to be independent programs or whether they become part of the linked program.[13] In this context, the following distinction can be made:

- System libraries

- Non-system libraries for GPL and AGPL applications

- Non-system libraries for applications licensed under other licenses than GPL and AGPL

Section 3 GPL-2.0 and section 1 (3) GPL-3.0 include exemptions from the license obligation to deliver "System Libraries" within the scope defined there, respectively.[14] Accordingly, license obligations for such system libraries need not be complied with if a Dockerfile contains the instruction to use these system libraries unmodified in a Docker container. The legal situation is therefore equivalent to the one that applies for base images (see above 4.3) and the essential role of the provider of the Dockerfile for the distribution is missing.

If, however, a Dockerfile specifies layers that are downloaded from third-party repositories and contain libraries (other than system libraries) which are intended for linking with a GPL-3.0 or AGPL-3.0 application in a Docker container, then the license obligations of GPL-3.0 or AGPL-3.0, respectively, must be complied with for these libraries for licensing reasons alone, for instance the source code must be provided (cf. section 1 GPL-3.0: "Corresponding Source includes ..., and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, ..."). This will apply for GPL-2.0 equivalently. Where the libraries are licensed under compatible licenses, the corresponding license conditions must also be complied as would be the case for an individual physical distribution. Accordingly, a Copyleft requirement cannot be circumvented by a decentralized distribution process.

---

11 For efforts of Red Hat to improve the situation see Peterson, S., ibid.
12 "However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance."
13 See for more details Jaeger, Till and Metzger, Aaxel, Open Source Software, 5th edition, 2020, 64 et seq; Meeker, Heather, Open Source for Business, A practical Guide to Open Source Software Licensing, 3rd edition 2020, 119 et seq; Working Paper on the legal implication of certain forms of
Software Interactions (a.k.a linking), Available at: <https://www.ifosslr.org/public/LinkingDocument.odt> [Accessed 16 February 2021].
14 The definition in section 1 GPL-3.0 reads as follows: 'The "System Libraries' of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A 'Major Component', in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it."

In all other cases in which program libraries are included as dependencies it can usually be assumed that the distribution is performed by the provider of the Dockerfile who has chosen these dependencies and who has the organizational control and therefore the essential role within the distribution process.

4.5 Updates

The handling of updates depends on whether the provider of the Dockerfile has control over the update. This can be answered in the affirmative if the provider itself (or someone on its behalf) uploads the update to a repository from where the recipient of the Dockerfile can subsequently obtain it. Whereas there is no distribution by the provider of the Dockerfile if the update is provided under the control of the operator of the repository (e.g. if the Dockerfile refers to the "latest version"). As in contrast to the program version named in the Dockerfile and chosen by the provider of the Dockerfile itself, they do not have relevant influence on the content of such updates.

4.6 Point in time to comply with license conditions

License obligations have to be complied with at the time of distribution (or communication to the public respectively). Since preparatory acts in a chain of distribution steps – such as the delivery of a Dockerfile – can already be considered a distribution, license obligations ought, strictly speaking, to be fulfilled at the time of delivery of the Dockerfile. It is, however, conceivable to interpret the Open Source licenses in such a way that it would be sufficient to comply with the license obligations at the time of download from a repository. This is supported by the fact that at the time of distribution of the Dockerfile it may still be unclear which program code in particular is included in a downloaded layer. This is the case, for example, if "latest" is specified for a program version.

However, as long as the license obligations are not fully met in the repositories involved, it is recommended to comply with the license obligations independently and to deliver a file with the required compulsory information (e.g. license texts, copyright notices, source code offer) along with the Dockerfile.

5. Conclusion

The responsibility for compliance with the license terms of FOSS licenses included in a Docker container cannot be circumvented by distributing a Dockerfile with the consequence that the recipient downloads the software directly from publicly accessible repositories. The analysis of the case law of the Court of Justice of the European Union reveals that such preparatory acts may also fall within the meaning of "distribution". Nevertheless, the peculiarities of the interaction of computer programs must be taken into account, which may entail that e.g. system requirements do not fall under the responsibility of the provider. However, repositories that offer Docker layer itself in a FOSS non-compliant way create also risks for the provider of Dockerfiles which reference to such repositories. FOSS license compliance thus becomes a shared responsibility between providers of Dockerfiles and public repositories for Docker layer.

About the author

Till Jaeger has been a partner at JBB Rechtsanwälte since 2001 (www.jbb.de). He is a Certified Copyright and Media Law Attorney and advises large and medium-sized IT businesses as well as government authorities and software developers on matters involving contracts, licensing and IP rights.

One particular focus of Till Jaeger's work is on the legal issues created by free and open source software (FOSS). He is co-founder of the Institute for Legal Aspects of Free & Open Source Software, ifrOSS (www.ifross.org), contributing to its work with academic publications, lectures and seminars in the fields of software law and copyright law.

Till Jaeger is a lecturer at the Humboldt University Berlin in the subjects of IT law and IP law and general counsel of Open Source Automation Development Lab (OSADL) eG.

He represented the gpl-violations.org project in several lawsuits to enforce the GPL and has published articles and books related to legal questions of Free and Open Source Software (among them Jaeger/Metzger, Open Source Software - Rechtliche Rahmenbedingungen der Freien Software, 5th ed. Munich 2020, and Van den Brande/Coughlan/Jaeger - The International FOSS Law Book, 2nd ed. Munich 2014). He was member of the Committee C in the GPLv3 drafting process.